Canadian Nuclear Safety Commission
Commission canadienne de sûreté nucléaire

REGULATORY GUIDE

Computer Programs Used in Design and Safety Analyses of Nuclear Power Plants and Research Reactors

G-149

October 2000

Canada

REGULATORY DOCUMENTS

The Canadian Nuclear Safety Commission (CNSC) operates within a legal framework that includes law and supporting regulatory documents. Law includes such legally enforceable instruments as acts, regulations, licences and orders. Regulatory documents such as policies, standards, guides, notices, procedures and information documents support and provide further information on these legally enforceable instruments. Together, law and regulatory documents form the framework for the regulatory activities of the CNSC.

The main classes of regulatory documents developed by the CNSC are:

Regulatory policy: a document that describes the philosophy, principles and fundamental factors used by the CNSC in its regulatory program.

Regulatory standard: a document that is suitable for use in compliance assessment and describes rules, characteristics or practices which the CNSC accepts as meeting the regulatory requirements.

Regulatory guide: a document that provides guidance or describes characteristics or practices that the CNSC recommends for meeting regulatory requirements or improving administrative effectiveness.

Regulatory notice: a document that provides case-specific guidance or information to alert licensees and others about significant health, safety or compliance issues that should be acted upon in a timely manner.

Regulatory procedure: a document that describes work processes that the CNSC follows to administer the regulatory requirements for which it is responsible.

Document types such as regulatory policies, standards, guides, notices and procedures do not create legally enforceable requirements. They support regulatory requirements found in regulations, licences and other legally enforceable instruments. However, where appropriate, a regulatory document may be made into a legally enforceable requirement by incorporation in a CNSC regulation, a licence or other legally enforceable instrument made pursuant to the *Nuclear Safety and Control Act*.

101-00801/

REGULATORY GUIDE

Computer Programs Used in Design and Safety Analyses of Nuclear Power Plants and Research Reactors

G-149

Published by the
Canadian Nuclear Safety Commission
October 2000

Computer Programs Used in Design and Safety Analyses of Nuclear Power Plants and Research Reactors
Regulatory Guide G-149

Published by the Canadian Nuclear Safety Commission

© Minister of Public Works and Government Services Canada 2000

Extracts from this document may be reproduced for individual use without permission provided the source is fully acknowledged. However, reproduction in whole or in part for purposes of resale or redistribution requires prior written permission from the Canadian Nuclear Safety Commission .

Catalogue number CC173-3/2-149E
ISBN 0-662-29526-9

Ce document est également disponible en français.

Document availability

The document can be viewed on the CNSC website. Copies in English or French may be ordered using the contact information below:

Communications Division
Canadian Nuclear Safety Commission
P.O. Box 1046, Station B
280 Slater Street
Ottawa, Ontario K1P 5S9
CANADA

Telephone: (613) 995-5894 or 1-800-668-5284 (Canada only)
Facsimile: (613) 992-2915
E-mail: info@cnsc-ccsn.gc.ca
Website: www.nuclearsafety.gc.ca

TABLE OF CONTENTS

1.0 PURPOSE 1

2.0 SCOPE 1

3.0 BACKGROUND 1
3.1 Regulatory framework 1
3.2 Licensing process 1
3.3 Relevant legislation 2
3.4 Commitment to quality 3

4.0 DEVELOPING A COMPUTER PROGRAM 3
4.1 Developing a computer program in phases 3
4.2 Development phases 3
4.3 Guidelines for verification and design review of development phases 3
4.4 Guidelines for validation 4
4.5 Overview of verification and design review 4
4.6 Verification 4
4.7 Verifying the requirements specifications 4
4.8 Verifying the program design 5
4.9 Verifying the coding 5
4.10 Verifying the program integration 5
4.11 Performing design review 6
4.12 Performing validation 7

5.0 MAINTENANCE 9
5.1 Configuration management 9
5.2 Change control 10

6.0 USING COMPUTER PROGRAMS 10
6.1 Overview 10
6.2 Uncertainty analysis 11
6.3 User prerequisites 11
6.4 User support 12
6.5 User options 12
6.6 Verifying the application process 12

7.0 DOCUMENTATION 13
7.1 Overview 13
7.2 Creating development documentation 13
7.3 Application documentation 14

8.0 EXISTING COMPUTER PROGRAMS 15

9.0 PROCUREMENT OF COMPUTER PROGRAMS 15

GLOSSARY 16

COMPUTER PROGRAMS USED IN DESIGN AND SAFETY ANALYSES OF NUCLEAR POWER PLANTS AND RESEARCH REACTORS

1.0 PURPOSE

This regulatory guide is intended to provide guidance to licensees involved in the development, maintenance and use of computer programs used in the design and safety analysis of nuclear power plants and research reactors so that a high degree of confidence may be placed in both the programs and the results of their application.

2.0 SCOPE

This guide applies to licensees whose computer programs are used in:

- designing or supporting the design of a nuclear power plant or research reactor
- analyzing operational transients, incidents or accidents.

This guide does not apply to operational control systems software.

For computer programs developed before the effective date of this regulatory guide, the degree of applicability is specified in section 8 of this guide.

3.0 BACKGROUND

3.1 Regulatory framework

The Canadian Nuclear Safety Commission (CNSC) is the federal agency that regulates the use of nuclear energy and materials to protect health, safety, security and the environment, and to respect Canada's international commitments on the peaceful use of nuclear energy.

The *Nuclear Safety and Control Act* ("the Act") requires persons or organizations to be licensed by the CNSC for carrying out the activities referred to in Section 26 of the Act, unless otherwise exempted. The associated regulations stipulate prerequisites for CNSC licensing, and the obligations of licensees and workers.

3.2 Licensing process

The CNSC typically applies a phased process to its licensing of nuclear facilities and activities. For major facilities, this process begins with a consideration of the environmental impacts of the proposed project, and proceeds progressively through site preparation, construction, operation, decommissioning and abandonment phases.

The *Nuclear Safety and Control Act* and regulations require licence applicants to provide certain information at each licensing stage. The type and level of detail of this information will vary to accommodate the licensing stage and specific circumstances.

At all licensing stages, applications may incorporate (directly or by reference) new or previously submitted information, in accordance with legislated requirements and the best judgement of the applicant. An application that is submitted at one licensing stage can become a building block for the next stage.

Upon receipt of an application that is complete, the CNSC reviews it to determine whether the applicant is qualified to carry on the proposed activity, and has made adequate provision for the protection of the environment, the health and safety of persons, and the maintenance of national security and measures required to implement international obligations to which Canada has agreed. If satisfied, the CNSC may issue, renew, amend or replace a licence that contains relevant conditions. Typically, this licence will incorporate the applicant's undertakings, and will contain other conditions that the CNSC considers necessary.

3.3 Relevant legislation

The CNSC uses this guide to assess information submitted as part of a licence application. Specifically, the *General Nuclear Safety and Control Regulations*, under paragraph 3(l)(*i*), requires that applicants provide "a description and the results of any test, analysis or calculation performed to substantiate the information included in the application." In addition, section 6 of the *Class I Nuclear Facilities Regulations* requires that the application for a licence to operate a Class 1 nuclear facility contains "(*b*) a description of the systems and equipment at the nuclear facility, including their design and their design operating conditions;" and "(*c*) a final safety analysis report demonstrating the adequacy of the design of the nuclear facility."

It is the licensee's responsibility to ensure that the computer programs used in design and safety analyses of nuclear power plants and research reactors and the output of these programs are reliable and adequate for their intended applications. These objectives can be attained either by:

(a) using the methods of development, maintenance and use of computer programs suggested in this guide, or
(b) applying other methods that are equally, or more, effective.

3.4 Commitment to quality

An organization's commitment to software quality is important, and it is the responsibility of licensees to develop a quality-assurance program for the development, maintenance and use of computer programs.

Adherence to relevant standards referred to in this guide, where appropriate, is one of the important elements in producing and maintaining high-quality computer programs.

4.0 DEVELOPING A COMPUTER PROGRAM

4.1 Developing a computer program in phases

Develop computer programs in phases, using the output of one phase to help set the requirements for one or more of the phases that follow. Typical development phases (see Glossary for definitions) include the following:

- problem definition
- requirements specification
- program design
- coding
- program integration
- validation

4.2 Development phases

Each development phase should consist of the following elements:

- define the input and requirement specifications — the functions to be performed and the expected outputs — based on the preceding phase;
- design, develop or test, as stated in the specifications;
- perform verification and design review, where appropriate, to ensure that the product matches the requirement specifications defined for the phase. If a phase cannot be verified until after more than one phase is complete, each phase should be verified iteratively.

4.3 Guidelines for verification and design review of development phases

With the exception of the validation phase, guidelines for the verification and design review of development phases are provided in sections 4.5 through 4.11 of this guide. Although there are no explicit guidelines for other development activities for these phases, the important elements to be considered in each development phase may be obtained from these guidelines.

4.4 Guidelines for validation

Guidelines for validation are provided in section 4.12 of this guide. Although validation has been described as a development phase, the activities of this phase may require extension beyond development phases to ensure that the computer program is validated for specific applications.

4.5 Overview of verification and design review

Perform verification and design review to show whether or not a program meets its requirements specifications. Specifically, these activities show the following:

- The mathematical equations adequately reflect the phenomena and processes of the physical system.
- The program design correctly reflects the mathematical equations.
- The program accurately reflects the design.
- The program's results satisfy its requirements specifications such that the results mirror the behaviour of the physical system.

Prepare a plan of the activities for verification and design review early during program development, for example, in the requirements specification phase. The plan will typically include the objectives, verification and design review approach, the schedule, and the project organization and management.

Review the verification and design review plan on a set schedule, updating the plan as necessary.

4.6 Verification

Assign verification tasks to the developer of a computer program and independent, qualified personnel.

Place the outputs to be verified under configuration control (refer to section 5 of this guide) before starting verification.

4.7 Verifying the requirements specifications

Demonstrate that:

- the specifications conform to the standards adopted for the program development so that, for example, all required sections are present and each section contains all required information;

- the specifications include the following detail:
  - all required functions;
  - all program input and output requirements, described in enough detail for a program to be designed;
  - identification of development standards;
  - acceptance criteria;
- the specifications are clear and unambiguous.

4.8 Verifying the program design

Demonstrate that:

- the design conforms to the standards adopted for the development of the program so that, for example, all required sections are present and each section contains all the required information;
- the design can be traced to the requirements specifications, so that all requirements are reflected in the design and all design features stem from these requirements;
- the design is complete, showing the required functions and specifying the program inputs and outputs, the operational environment and the processing steps;
- the design is internally consistent;
- the design is clear and unambiguous.

4.9 Verifying the coding

Demonstrate that the source program:

- conforms to programming and language standards adopted for the development of the computer program;
- does not include unnecessarily complex coding;
- does include error-checking capabilities;
- does include logic that is consistent with the design specifications.

4.10 Verifying the program integration

Demonstrate that:

- the integrated program conforms with the requirements of the operating system;
- the integrated program interfaces properly with external files;
- the program links correctly — for example, all modules and subroutines are properly linked;
- the control language is correct;
- the processing and transmission of data between modules are correct.

4.11 Performing design review

Perform a design review in at least the following phases:

- requirements specifications
- program design

Include representatives of the following groups in the design review team:

- developers
- programmers
- end-users
- a quality-assurance group

The design review team should include independent members with well-recognized expertise in review areas, for example, experts in the physical sciences, mathematics, computer science and software programming.

In the design review of the requirement specifications, demonstrate that:

- the technical description of the problem to be solved is correct, complete and consistent with the statement of the problem;
- the solution methods are appropriate, based on state-of-the-art knowledge.

During the design review of the requirement specifications, review the following:

- the scope of the problem, the specifications of the physical system and the identification of the initial and boundary conditions relevant to the phenomena to be modelled;
- knowledge of the physical system;
- identification and understanding of the phenomena;
- the modelling concept and the models of the physical system and phenomena, including assumptions, methods of approximation and simplification;
- the limitations of the models, and what these limitations imply;
- the analytical and numerical solution methods, including the numerical techniques and algorithms;
- the requirement specifications and instructions for coding the computer program.

The review of the computer program design aims to demonstrate that all program requirements are implemented correctly in the design, that all aspects of the design can be traced to the requirements and that the design is feasible, consistent and based on state-of-the-art knowledge.

During the design review of the computer program, review the following:

- program conceptual design
- basic logic
- system flow diagrams
- numerical techniques and algorithms
- operational environment and program interface

In the design review of the computer program, demonstrate that:

- the numerical techniques are appropriate for the types of problems to be solved, as well as for the treatment of boundary or inter-phase conditions and special situations, such as singularities;
- the mathematical equations have been transformed correctly into a numeric scheme of solutions, and the steps in the solution design are in the correct sequence;
- the user options and their restrictions are clearly described.

4.12 Performing validation

Perform validation to determine the accuracy of the computer program's predictions, and to help determine the uncertainties discussed in section 6.2 of this guide.

Develop a validation plan early in the development of a computer program. The validation plan should include:

- the objectives and scope of the validation;
- the validation approach, including the method of validation, requirements to be validated, acceptance criteria for each requirement, and method for evaluation of validation results;
- the basis for the selection of validation cases, and the specifications for each case;
- the validation database;
- the validation procedure, including the validation result reporting procedure.

Review the validation plan on a set schedule, updating the plan as necessary.

Assign program validation tasks to qualified persons, specifically developers and those who are familiar with the program, the sources of the data being used to validate the program and the program's intended use.

Validation tests are meant to determine:

- a program's accuracy;
- how well a program and its conceptual models reflect physical processes or systems;
- a program's capabilities and limits, including the range of parameters to which the program may properly be applied.

During validation, systematically test the performance of conceptual models, empirical correlations and the integrated program. Relevant are tests against analytical solutions, operational data, and data from separate effects and integral effect tests. Comparisons against validated programs may be included.

To perform systematic testing:

- develop a test plan;
- identify and rank important phenomena and parameters;
- identify the models, empirical correlations or components of the program that are to be tested;
- identify suitable existing experimental data, as well as data to be obtained from new experiments;
- assess the program using separate effect and integral effect tests, operational measurements, analytical solutions or the results of one or more validated programs;
- assess the sensitivity to the input options;
- evaluate validation results.

During validation, examine the following key areas when applicable:

- the assumptions made to simplify the physical system;
- the theoretical and experimental bases for the models and empirical correlations, as well as their applicability range;
- the compatibility of the ranges of the parameters, as well as the geometrical and phenomenological similarities between the system being simulated and the experiments;
- the ability of the code to predict behaviour during integrated tests, to show that (1) no unwarranted interaction occurs between various models, and (2) the code accurately simulates how the system's parts interact;
- the sensitivity of individual models and empirical correlations, as well as the integrated code, to variations in key parameters or factors, particularly near the end of their allowable ranges and near-critical values such as singularities or discontinuity;

- potential errors arising from the use of the models and empirical correlations outside their intended ranges;
- the variation in code predictions when using various options of the alternative models and empirical correlations as recommended in the code documentation (refer to section 7 of this guide).

Note all validation results, clearly identifying the program's abilities, limits and ranges of applicability.

During validation, identify improvements that should be made to the program. Identify, as well, any need to enhance the validation data, including experimental data.

Write a report, based on validation results, that shows the correct and appropriate use of the program.

Verify that the validation process has been conducted in accordance with the validation plan, and that the validation results are accurately reported.

5.0 MAINTENANCE

5.1 Configuration management

Establish procedures for computer program maintenance and change control.

Adopt practices that protect program integrity. Clearly define the roles and responsibilities of those who are responsible for a program's integrity, as well as those who maintain and use it.

Maintain a configuration management system to help ensure program integrity and to keep track of a program's versions and components.

Typical components of a program to be configured include:

- source program
- object program
- executable program
- processors and operating systems
- files that control the generating of executable code from source code
- input data files
- documentation
- tools that support program development or maintenance

Configuration management involves:

- using a naming convention to uniquely identify the functional and physical characteristics of each configuration component
- controlling changes to the configuration components
- recording and reporting change processing and implementation status
- verifying compliance with configuration requirements

5.2 Change control

Set up a change control system to control changes to each configuration item. A change control system should include:

- a designated person responsible for all changes to a computer program;
- a change approval group whose members represent developers, users and those who performed the program's verification and validation;
- procedures to:
  - review proposed changes to a configuration item;
  - approve changes;
  - implement changes;
  - document changes and their rationales;
  - verify that a change was implemented correctly;
  - assess the impact of each change on the use of the program and on the quality of its predictions;
  - make any necessary revisions to existing documentation.

6.0 USING COMPUTER PROGRAMS

6.1 Overview

Establish guidelines and procedures for the use of computer programs. These should include guidelines for user prerequisites and the designation of persons responsible for ensuring the correct and appropriate use of computer programs.

Apply a program only to a problem or set of problems for which the program has been validated and designed to solve (only within the range of applicability identified in the verification and validation reports). If the program is used outside its range of validation, the validity of the extrapolation should be justified.

Minimize as far as possible the program's user effects. An illustration of user effects is: if two experienced users, with the same version of a computer program, perform a test or analyze a problem and obtain significantly different results, then the program may be user-dependent. The differences can be attributed to the use of input options, such as those for time step control, nodalization schemes, correlations, coefficients, or parameters.

Report errors and deficiencies in a program or its application to the person who is responsible for the maintenance of the program. Correct the errors and deficiencies, using the change control procedures listed in section 5.2 of this guide.

6.2 Uncertainty analysis

Evaluate a computer program's results to determine the uncertainties in each type of application of the program.

Account for potential uncertainties identified in the development and use of the program, including those due to:

- simplifications and assumptions made to compensate for lack of knowledge or to render a problem solvable, including assumptions that simplify equations, empirical correlations, and physical and system models;
- inadequate knowledge of the problem that the computer program will be used to analyze, such as complex phenomena, scaling effects, and initial and boundary conditions;
- limits to the ability to represent a physical system, such as those presented by nodalization techniques;
- inadequacy of the data used to validate the program, for example, instrumentation errors and test repeatability;
- compensating errors or uncertainties;
- propagation of uncertainties, i.e. the uncertainty of the results increases as the transient progresses.

6.3 User prerequisites

Take steps to ensure that each user of a computer program possesses experience that matches or exceeds the safety profile of the program's application.

Important factors in the knowledge and experience needed by the program's user are:

- good knowledge and understanding of the information in all application documents and the verification and design review reports, as well as familiarity with other information in the design documents;
- sufficient experience in using the program for the intended application, and similar ones, with a good knowledge of program responses to (1) various system and phenomenon models and to (2) assumptions and changes in important parameters;
- extensive knowledge of the problem to be analyzed, for example, knowledge of the physical system and the phenomena being modelled.

6.4 User support

User support should be available to advise the user on the correct use of a computer program and proper modellings or simulations for important or large and complex programs, such as thermal hydraulics system programs or multi-field coupling programs.

Set up a user support team if warranted, and include the following members:

- the developer or those who know the program well;
- those who performed, or who know a great deal about, the program's verification and validation;
- those who have a thorough knowledge of the system to be modelled and the problem to be solved.

6.5 User options

User input options should be avoided as far as practicable.

Write guidelines for each user option, and make them available to users.

6.6 Verifying the application process

Verify that the application process is correct by demonstrating that it satisfies the guidelines for the use of computer programs provided in this section.

If an application process cannot be satisfactorily verified, identify exceptions, and provide a justification for each.

7.0 DOCUMENTATION

7.1 Overview

The documentation of a computer program and its associated components should conform with documentation standards adopted in the development of the program. Verify the program documents to demonstrate that they are complete, consistent, clear and unambiguous.

7.2 Creating development documentation

In the development documentation include:

- problem definition
- program development plan
- requirements specifications
- design specification
- programmer's manual
- verification and design review report

In the problem definition, document in detail the statement of the problem and the rationale and objectives of the program.

In the program development plan, describe the organization, schedule and activities related to the development, design review, verification and validation of the program. Also document procedures for updating the program development plan.

In the requirements specifications, clearly identify all program requirements. In addition, provide the basis for verifying the program design and evaluating the performance of the program through validation.

In the design specification, specify the elements needed for program coding, including the logical structure, information flow, models, numerical solution techniques, discretization method, data structure, supporting software and hardware, and the operating environment.

In the programmer's manual, describe the program flow and structure, the method of translating theory into coding, instructions for maintaining and modifying the computer program and conventions on programming practices, such as variable naming and program commenting.

In the verification and design review report, document verification activities, including the description of the method and results of each verification or design review, the range of applicability and recommendations to users regarding the capability and limits of the computer program. Also include the input data used in each verification.

7.3 Application documentation

Application documentation should include:

- program abstract
- theory manual
- user's manual, including specific user guidelines
- one or more validation reports
- source code
- sample input and output

In the program abstract, provide the following information about the program:

- the name and version of the program and applicable configuration items;
- the program's purpose, capabilities, limitations and operating environment;
- a summary of the problem(s) the program is designed to solve;
- the names of the organization and key individuals responsible for code development, support and maintenance of the program.

In the theory manual, describe the basis of the computer program including:

- the physical systems to be modelled and their models;
- the phenomena to be modelled and their models or empirical correlations;
- mathematical formulations of the problem and solution techniques;
- assumptions and constraints, and what they imply about the limits of the program's capabilities and the range of its applicability;
- references.

In the user's manual include enough information to run the computer program. Provide details on the required input data, techniques for restarting the code, diagnostic information and options available to the user, including:

- basic information on compiling, linking and executing the code;
- detailed description of all input parameters indicating type and format;
- a discussion of pre- and post-processing of the code, including code restarts;
- sample input and output files that show representative problems;
- a description of possible execution error messages and termination messages.

In addition, to minimize the risk of inappropriate use of a program, provide user guidelines on the following topics:

- the correct and proper use of the program for each of its applications;
- the range of applicability;
- the program's capabilities and limitations;
- the options available to users, such as model options or choice of nodalization schemes.

In the validation report, include the following:

- validation plan;
- validation results;
- evaluation of the validation results, including a determination of the computer program's accuracy and range of uncertainty for each test;
- recommendations for improvement to the program;
- specification of further experiments;
- recommendations for the correct and appropriate use of the program for each application, range of applicability, and program capability and limitations.

8.0 EXISTING COMPUTER PROGRAMS

Existing computer programs are those used by the CNSC licensee before the effective date of this document.

Assess and document the extent to which existing programs conform with this guide.

Improve, to the extent practicable, the areas that do not conform with relevant sections of this guide so that there is an increase in the level of confidence in a program consistent with the program's impact on safety.

Provide justification for any non-conformance with the guidelines in this guide.

9.0 PROCUREMENT OF COMPUTER PROGRAMS

Establish procedures for the procurement and maintenance of computer programs. Ensure that all computer programs destined for use by a CNSC licensee, but developed outside a CNSC licensee's organization, meet or exceed the criteria of this guide.

GLOSSARY

coding
The step in which a computer program design is transformed into a sequence of machine-readable instructions suitable for processing by a computer.

computer program
A sequence of machine-readable instructions suitable for processing by a computer; also referred to in this document as a program.

developer
A person who writes the code for a computer program, or a portion of one, and who works on its associated documents.

discretization
A method of approximation of the true mathematical function to be integrated.

problem definition
A detailed description of the problem and the rationale for the decision to develop a computer program.

program design
The elements required for program coding, including the logical structure, information flow, models, numerical solution techniques, discretization method, data structure, supporting software and hardware, operating environment, and other features required to satisfy the requirements specifications.

program integration
The step in which a source program is integrated with other components, such as library routines, system linking specifications, operating system control language, external data libraries and hardware environment to form an integrated, executable program.

requirements specifications
The requirements the program must satisfy and the basis for verifying the program design and then evaluating the program through validation.

source program
A program that must be compiled, assembled or interpreted before it can be executed.

user
A person who uses a computer program.

validation
A process by which the results of a computer program are compared with operational measurements, experimental data, or analytical solutions, to determine the program's accuracy and uncertainty.

verification
The step in which it is determined whether the products of a given phase of the program development cycle fulfil the requirements set during the previous phase.

Commission canadienne de sûreté nucléaire
Canadian Nuclear Safety Commission

GUIDE D'APPLICATION DE LA RÉGLEMENTATION

Les programmes informatiques utilisés lors de la conception et des analyses de sûreté des centrales nucléaires et des réacteurs de recherche

G-149

Octobre 2000

Canada

DOCUMENTS D'APPLICATION DE LA RÉGLEMENTATION

La Commission canadienne de sûreté nucléaire (CCSN) fonctionne à l'intérieur d'un cadre juridique constitué de la législation et, à l'appui, de documents d'application de la réglementation. Le terme « législation » renvoie à différents instruments légaux exécutoires : des lois, des règlements, des permis et des ordres. Quant aux documents d'application de la réglementation — des politiques, des normes, des guides, des avis, des procédures et des documents d'information —, ils soutiennent et expliquent davantage ces instruments. Les activités de réglementation de la CCSN reposent sur ces instruments et ces documents.

Les documents d'application de la réglementation de la CCSN relèvent des principales classes suivantes :

Politique d'application de la réglementation : un document qui décrit la doctrine, les principes et les facteurs fondamentaux utilisés par la CCSN dans son programme de réglementation.

Norme d'application de la réglementation : un document qui peut servir à une évaluation de conformité et qui décrit les règles, les caractéristiques ou les pratiques que la CCSN accepte comme conformes aux exigences réglementaires.

Guide d'application de la réglementation : un document qui sert de guide ou qui décrit des caractéristiques ou des pratiques recommandées par la CCSN et qui, d'après elle, permettent de respecter les exigences réglementaires ou d'améliorer l'efficacité administrative.

Avis d'application de la réglementation : un document qui contient des conseils et des renseignements propres à un cas donné et qui sert à alerter les titulaires de permis et d'autres personnes à propos d'importantes questions de santé, de sûreté ou de conformité auxquelles il faut donner suite en temps utile.

Procédure d'application de la réglementation : un document qui décrit les modalités de travail qu'utilise la CCSN pour administrer les exigences réglementaires dont elle est responsable.

Les politiques, normes, guides, avis et procédures d'application de la réglementation ne créent pas d'exigences exécutoires, mais étayent les exigences réglementaires des règlements, des permis et des autres instruments exécutoires. Néanmoins, le cas échéant, un document d'application de la réglementation peut être transformé en instrument exécutoire par son incorporation dans un règlement de la CCSN, dans un des permis qu'elle délivre ou dans un autre instrument exécutoire établi en vertu de la *Loi sur la sûreté et la réglementation nucléaires*.

GUIDE D'APPLICATION DE LA RÉGLEMENTATION

Les programmes informatiques utilisés lors de la conception et des analyses de sûreté des centrales nucléaires et des réacteurs de recherche

G-149

Publié par la
Commission canadienne de sûreté nucléaire
Octobre 2000

Les programmes informatiques utilisés lors de la conception et des analyses de sûreté des centrales nucléaires et des réacteurs de recherche
Guide d'application de la réglementation G-149

Publié par la Commission canadienne de sûreté nucléaire

© Ministre des Travaux publics et Services gouvernementaux Canada 2000

La reproduction d'extraits du présent document à des fins personnelles est autorisée à condition d'en indiquer la source en entier. Toutefois, sa reproduction en tout ou en partie à des fins commerciales ou de redistribution nécessite l'obtention préalable d'une autorisation écrite de la Commission canadienne de sûreté nucléaire.

N° de cat. CC173-3/2-149F
ISBN 0-662-85192-7

This document is also available in English.

Disponibilité du présent document

Les personnes intéressées pourront consulter le présent document sur le site Web de la CCSN ou en commander des exemplaires, en français ou en anglais, en communiquant avec la :

Division des communications
Commission canadienne de sûreté nucléaire
Case postale 1046, Succursale B
280, rue Slater
Ottawa (Ontario) K1P 5S9
CANADA

Téléphone : (613) 995-5894 ou 1-800-668-5284 (au Canada)
Télécopieur : (613) 992-2915
Courriel : info@cnsc-ccsn.gc.ca
Site Web : www.suretenucleaire.gc.ca

TABLE DES MATIÈRES

1.0 OBJET 1

2.0 PORTÉE 1

3.0 CONTEXTE 1
3.1 Cadre de réglementation 1
3.2 Processus de délivrance de permis 2
3.3 Dispositions législatives applicables 2
3.4 Engagement à l'égard de la qualité 3

4.0 DÉVELOPPEMENT DES PROGRAMMES INFORMATIQUES 3
4.1 Développement des programmes informatiques par étapes 3
4.2 Étapes de développement 4
4.3 Lignes directrices pour la vérification et la revue de la conception relatives aux étapes de développement 4
4.4 Lignes directrices pour la validation 4
4.5 Aperçu de la vérification et de la revue de la conception 4
4.6 Vérification 5
4.7 Vérification de la spécification des exigences 5
4.8 Vérification de la conception du programme informatique 5
4.9 Vérification du codage 6
4.10 Vérification de l'intégration du programme informatique 6
4.11 Exécution de la revue de la conception 6
4.12 Exécution de la validation 8

5.0 MAINTENANCE 10
5.1 Gestion de la configuration 10
5.2 Contrôle des changements 11

6.0 UTILISATION DES PROGRAMMES INFORMATIQUES 11
6.1 Aperçu 11
6.2 Analyse des incertitudes 12
6.3 Préalables exigés de l'utilisateur 13
6.4 Soutien de l'utilisateur 13
6.5 Options de l'utilisateur 13
6.6 Vérification du processus d'application 14

7.0 DOCUMENTATION 14
7.1 Aperçu 14
7.2 Création de la documentation de développement 14
7.3 Documentation d'application 15

8.0 PROGRAMMES INFORMATIQUES EXISTANTS 17

9.0 ACQUISITION DE PROGRAMMES INFORMATIQUES 17

GLOSSAIRE 18

LES PROGRAMMES INFORMATIQUES UTILISÉS LORS DE LA CONCEPTION ET DES ANALYSES DE SÛRETÉ DES CENTRALES NUCLÉAIRES ET DES RÉACTEURS DE RECHERCHE

1.0 OBJET

Le présent guide d'application de la réglementation fournit des orientations aux titulaires de permis qui sont appelés à développer, à maintenir et à utiliser des programmes informatiques servant à la conception et aux analyses de sûreté des centrales nucléaires et des réacteurs de recherche. Il vise à faire en sorte que l'on puisse avoir un haut niveau de confiance tant dans ces programmes eux-mêmes que dans les résultats obtenus lors de leur mise en application.

2.0 PORTÉE

Le présent guide est destiné aux titulaires de permis qui utilisent des programmes informatiques créés pour :

- la conception ou le soutien à la conception des centrales nucléaires et des réacteurs de recherche;
- l'analyse de phénomènes transitoires, d'incidents ou d'accidents.

Il ne s'applique pas aux logiciels des systèmes de contrôle opérationnel.

Pour ce qui est des programmes élaborés avant l'entrée en vigueur du présent guide, leur degré d'applicabilité est précisé à la section 8 du présent guide.

3.0 CONTEXTE

3.1 Cadre de réglementation

La Commission canadienne de sûreté nucléaire (CCSN) est l'organisme fédéral qui réglemente l'utilisation de l'énergie et des matières nucléaires afin de protéger la santé, la sûreté, la sécurité et l'environnement et de respecter les engagements internationaux du Canada à l'égard de l'utilisation pacifique de l'énergie nucléaire.

La *Loi sur la sûreté et la réglementation nucléaires* (« la Loi ») exige des personnes ou des organisations qu'elles détiennent un permis de la CCSN avant de réaliser les activités décrites à l'article 26 de la Loi, à moins d'en être exemptées. Les règlements d'application de la Loi stipulent des préalables pour la délivrance d'un permis de la CCSN, ainsi que les obligations des titulaires de permis et des travailleurs.

3.2 Processus de délivrance de permis

De façon générale, la CCSN applique un processus par étapes pour délivrer les permis applicables aux installations et aux activités nucléaires. Quand il s'agit d'installations importantes, la démarche commence par un examen des incidences environnementales du projet, puis s'oriente progressivement vers les phases subséquentes : préparation du site, construction, exploitation, déclassement et abandon.

En vertu de la *Loi sur la sûreté et la réglementation nucléaires* et de ses règlements, le demandeur de permis doit communiquer certains renseignements à chaque étape du processus de délivrance de permis. Le genre et le degré de détail des renseignements varient en fonction de l'étape et des circonstances.

À toutes les étapes du processus, le demandeur peut communiquer des renseignements nouveaux ou déjà présentés (directement ou par renvoi), conformément aux exigences législatives et selon son bon jugement. Une demande présentée à une étape particulière peut devenir une composante de l'étape suivante.

Quand elle reçoit une demande complète, la CCSN l'examine afin de déterminer si le demandeur est qualifié pour exécuter l'activité proposée. Elle vérifie également si le demandeur a pris des dispositions suffisantes pour protéger l'environnement, ainsi que la santé et la sécurité des personnes, pour assurer le maintien de la sécurité nationale et pour appliquer les mesures requises de manière à se conformer aux obligations internationales du Canada. Si elle juge que le demandeur satisfait à ces critères, la CCSN peut délivrer, renouveler, modifier ou remplacer un permis assorti des conditions pertinentes. Habituellement, le permis énumère les engagements du demandeur et d'autres conditions jugées nécessaires par la CCSN.

3.3 Dispositions législatives applicables

La CCSN utilise le présent guide pour évaluer les renseignements fournis dans la demande de permis. De façon plus précise, le *Règlement général sur la sûreté et la réglementation nucléaires* exige, aux termes de l'alinéa 3 (l) *i*), que le demandeur de permis fournisse « une description et les résultats des épreuves, analyses ou calculs effectués pour corroborer les renseignements compris dans la demande ». En outre, à l'article 6 du *Règlement sur les installations nucléaires de catégorie I*, il est précisé que la demande de permis pour exploiter une installation nucléaire de catégorie I doit notamment comprendre « *b*) une description des systèmes et de l'équipement de l'installation nucléaire, y compris leur conception et leurs conditions nominales de fonctionnement; » et « *c*) un rapport final d'analyse de la sûreté démontrant que la conception de l'installation nucléaire est adéquate; ».

Les titulaires de permis ont la responsabilité de veiller à ce que tant les programmes informatiques servant à la conception et aux analyses de sûreté des centrales nucléaires et des réacteurs de recherche que les résultats obtenus lors de leur mise en application demeurent fiables et conformes au but recherché. Ces objectifs peuvent être atteints en ayant recours :

a) soit aux méthodes de développement, de maintenance et d'utilisation proposées dans le présent guide;
b) soit à d'autres méthodes tout aussi efficaces, voire plus efficaces encore.

3.4 Engagement à l'égard de la qualité

L'engagement d'une organisation à l'égard de la qualité de ses logiciels constitue un élément important. Les titulaires de permis ont la responsabilité d'élaborer un programme d'assurance de la qualité pour le développement, la maintenance et l'utilisation des programmes informatiques.

Le respect, le cas échéant, des normes pertinentes mentionnées dans le présent guide représente également un facteur important pour assurer la production et la maintenance de programmes informatiques de haute qualité.

4.0 DÉVELOPPEMENT DES PROGRAMMES INFORMATIQUES

4.1 Développement des programmes informatiques par étapes

Développer les programmes informatiques en plusieurs étapes, les résultats obtenus au terme de chacune contribuant à établir les exigences qui seront recherchées lors d'une ou de plusieurs étapes subséquentes. Ce processus comporte habituellement les étapes suivantes, qui sont définies dans le Glossaire du présent guide :

- la définition du problème;
- la spécification des exigences;
- la conception du programme;
- le codage;
- l'intégration du programme;
- la validation.

4.2 Étapes de développement

Chaque étape de développement devrait comporter les éléments suivants :

- en se fondant sur l'étape précédente, définir les entrées et établir la spécification des exigences — les fonctions à exécuter et les résultats attendus;
- concevoir, développer ou mettre à l'essai le produit selon la spécification des exigences qui a été établie;
- procéder à la vérification et à la revue de la conception, le cas échéant, pour s'assurer que le produit correspond effectivement à la spécification des exigences définie pour l'étape; si l'étape ne peut faire l'objet d'une vérification avant l'achèvement de plusieurs autres étapes, procéder itérativement à la vérification de chacune des étapes.

4.3 Lignes directrices pour la vérification et la revue de la conception relatives aux étapes de développement

Les lignes directrices pour la vérification et la revue de la conception relatives aux étapes de développement — à l'exclusion de l'étape de validation — sont énoncées dans les sections 4.5 à 4.11 du présent guide. Même s'il n'existe pas de lignes directrices explicites pour les autres activités de développement relevant de ces étapes, il est fait état dans les sections susmentionnées des éléments importants qui devront être pris en compte lors de chaque étape.

4.4 Lignes directrices pour la validation

Les lignes directrices pour la validation sont énoncées dans la section 4.12 du présent guide. Même si la validation a été intégrée aux étapes de développement, les activités s'y rapportant pourront se prolonger au-delà des étapes de développement pour s'assurer que le programme informatique est validé aux fins d'applications particulières.

4.5 Aperçu de la vérification et de la revue de la conception

Procéder à la vérification et à la revue de la conception pour établir si le programme informatique correspond ou non à la spécification des exigences. Plus particulièrement, ces activités viseront à démontrer que :

- les équations mathématiques rendent compte adéquatement des phénomènes et des processus du système physique;
- la conception du programme correspond effectivement aux équations mathématiques;
- le programme correspond effectivement à la conception;

- les résultats du programme correspondent à la spécification des exigences, et rendent ainsi compte le plus exactement possible du comportement du système physique.

Préparer le plan des activités de vérification et de revue de la conception dès le début du processus de développement du programme informatique — lors de l'étape de la spécification des exigences, par exemple. Habituellement, ce plan fera notamment état des objectifs poursuivis et de la démarche de vérification et de revue de la conception à adopter, ainsi que de l'échéancier, de l'organisation et de la gestion du projet.

Réviser, suivant un échéancier établi, le plan de la vérification et de la revue de la conception et le mettre à jour au besoin.

4.6 Vérification

Confier les activités de vérification du programme informatique au développeur et à du personnel qualifié indépendant.

Placer les sorties à vérifier sous le contrôle de la configuration (voir la section 5 du présent guide) avant d'amorcer la vérification.

4.7 Vérification de la spécification des exigences

Démontrer que la spécification des exigences :

- respecte les normes adoptées pour le développement du programme informatique, de sorte que l'on y reprend, par exemple, toutes les sections exigées, et que chaque section contient toute l'information voulue;
- comporte les éléments suivants :
  - toutes les fonctions requises;
  - toutes les exigences relatives aux entrées et aux sorties du programme, décrites suffisamment en détail pour permettre la conception du programme;
  - l'identification des normes de développement;
  - les critères d'acceptation;
- est claire et sans ambiguïté.

4.8 Vérification de la conception du programme informatique

Démontrer que la conception du programme informatique :

- respecte les normes adoptées pour le développement du programme, de sorte que l'on y reprend, par exemple, toutes les sections exigées, et que chaque section contient toute l'information voulue;

- peut être rattachée à la spécification des exigences, de sorte qu'elle respecte toutes les exigences relatives à la conception et que toutes les caractéristiques de conception découlent de ces exigences;
- est complète, de sorte qu'elle décrit non seulement les fonctions requises mais encore les entrées et les sorties du programme, l'environnement opérationnel et les étapes de traitement;
- est cohérente d'une section à l'autre;
- est claire et sans ambiguïté.

4.9 Vérification du codage

Démontrer que le programme source :

- respecte les normes de programmation et de langage adoptées pour le développement du programme;
- ne comprend pas de codage inutilement complexe;
- peut détecter les erreurs;
- comporte une logique correspondant à la spécification des exigences de conception.

4.10 Vérification de l'intégration du programme informatique

Démontrer que le programme informatique intégré :

- est conforme aux caractéristiques du système d'exploitation;
- interface de façon appropriée avec les fichiers externes;
- établit correctement des liens — tous les modules et sous-programmes sont reliés de façon appropriée, par exemple;
- utilise un langage de commande correct;
- traite et transmet correctement les données entre les modules.

4.11 Exécution de la revue de la conception

Procéder à une revue de la conception au moins lors des étapes suivantes :

- la spécification des exigences;
- la conception du programme informatique.

Inclure dans l'équipe de revue de la conception des représentants des groupes suivants :

- les développeurs;
- les programmeurs;
- les utilisateurs finaux;
- les membres de l'équipe d'assurance de la qualité.

Intégrer à l'équipe de revue de la conception des membres indépendants qui ont une compétence reconnue dans les domaines à l'étude — des experts dans les sciences physiques, en mathématique, en informatique et en programmation logicielle, par exemple.

Au cours de la revue de la conception visant la spécification des exigences, démontrer que :

- la description technique du problème à résoudre est correcte, complète et conforme à l'énoncé du problème;
- les méthodes envisagées pour le résoudre sont appropriées et fondées sur les connaissances les plus récentes.

Au cours de la revue de la conception visant la spécification des exigences, vérifier les éléments suivants :

- la portée du problème, la spécification des exigences relatives au système physique et l'identification des conditions, initiales et aux limites, du phénomène à modéliser;
- la connaissance du système physique;
- l'identification et la compréhension du phénomène;
- le concept de modélisation et les modèles du système physique et des phénomènes, y compris les hypothèses et les méthodes d'approximation et de simplification;
- les limitations inhérentes aux modèles et leurs incidences;
- les techniques de solution analytiques et numériques, y compris les algorithmes et techniques numériques;
- la spécification des exigences et les instructions de codage du programme informatique.

La revue de la conception du programme informatique a pour but de démontrer que toutes les exigences du programme sont correctement mises en œuvre dans la conception, que tous les aspects de la conception peuvent être rattachés à la spécification des exigences et que la conception est faisable, cohérente et fondée sur les connaissances les plus récentes.

Au cours de la revue de la conception du programme informatique, vérifier les éléments suivants :

- l'étude conceptuelle du programme;
- la logique de base;
- les organigrammes du système;
- les algorithmes et techniques numériques;
- l'environnement opérationnel et l'interface du programme.

Au cours de la revue de la conception du programme informatique, démontrer que :

- les techniques numériques sont appropriées pour les types de problèmes à résoudre, ainsi que pour le traitement des conditions aux limites ou entre les étapes et des situations spéciales — les singularités, par exemple;
- les équations mathématiques ont été transformées correctement en un schème numérique de solutions et les étapes de conception de la solution sont dans la séquence appropriée;
- les options de l'utilisateur, de même que leurs restrictions, sont clairement décrites.

4.12 Exécution de la validation

Procéder à une validation pour déterminer l'exactitude des prédictions du programme informatique et pour aider à déterminer les incertitudes abordées dans la section 6.2 du présent guide.

Élaborer un plan de validation dès le début du processus de développement du programme informatique. Inclure dans le plan de validation :

- les objectifs et la portée de la validation;
- la démarche de validation, y compris la méthode de validation, les exigences à valider, les critères d'acceptation pour chacune de ces exigences et la méthode d'évaluation des résultats de la validation;
- le fondement présidant au choix des cas de validation et la spécification des exigences rattachées à chacun de ces cas;
- la base de données de validation;
- la procédure de validation, y compris la procédure de compte rendu des résultats de la validation.

Réviser, suivant un échéancier établi, le plan de validation et le mettre à jour au besoin.

Confier les activités de validation du programme informatique à du personnel qualifié, et plus particulièrement aux développeurs et aux personnes qui connaissent le programme, les sources des données utilisées pour la validation et l'usage prévu.

Les essais de validation visent à vérifier :

- l'exactitude du programme informatique;
- le niveau suivant lequel le programm et ses modèles conceptuels réussissent à rendre compte des processus ou systèmes physiques;
- les capacités et limitations du programme, y compris ses domaines d'applicabilité.

Dans le cours de la validation, il faut mettre systématiquement à l'essai la performance des modèles conceptuels, des corrélations empiriques et du programme intégré. Au nombre des essais pertinents, figurent ceux qui portent sur les solutions analytiques, sur les données opérationnelles et sur les données résultant d'essais axés sur des effets distincts et sur l'effet intégral. Le programme peut en outre être comparé à des programmes déjà validés.

Pour effectuer de tels essais systématiques, il faut :

- établir un plan d'essais;
- identifier et classer les phénomènes et les paramètres importants;
- identifier les modèles, les corrélations empiriques ou les éléments du programme informatique qui doivent faire l'objet d'essais;
- identifier les données expérimentales existantes appropriées, ainsi que les données qui doivent provenir de nouvelles expériences;
- évaluer le programme grâce à des essais portant sur des effets distincts et sur l'effet intégral, à des mesures opérationnelles, à des solutions analytiques ou aux résultats d'un ou de plusieurs programmes déjà validés;
- évaluer la sensibilité aux options d'entrée;
- évaluer les résultats de la validation.

Dans le cours de la validation, examiner les éléments clés suivants, le cas échéant :

- les hypothèses qui ont été posées pour simplifier le système physique;
- les bases théoriques et expérimentales des modèles et des corrélations empiriques, ainsi que leurs domaines d'applicabilité;
- la compatibilité des gammes de paramètres, ainsi que les ressemblances géométriques et phénoménologiques entre le système simulé et les expériences;
- la capacité du code à prédire le comportement pendant les essais intégrés, en montrant (1) qu'il n'y a aucune interaction injustifiée entre les divers modèles et (2) que le code arrive effectivement à simuler avec précision la façon dont les éléments du système interagissent entre eux;
- la sensibilité des modèles individuels et des corrélations empiriques, ainsi que du code intégré, aux variations de paramètres ou de facteurs clés, particulièrement près des limites de leurs gammes admissibles et près des valeurs critiques — celles des singularités ou des discontinuités, par exemple;
- les erreurs éventuelles dues à l'utilisation de modèles et de corrélations empiriques au-delà des limites prévues;
- la variation dans les prédictions du code lorsque sont utilisées diverses options relatives aux corrélations empiriques et modèles de substitution recommandées dans la documentation du code (voir la section 7 du présent guide).

Noter tous les résultats de la validation, et identifier clairement les capacités, les limitations et les domaines d'applicabilité du programme informatique.

Dans le cours de la validation, identifier les améliorations qui doivent être apportées au programme informatique. Déterminer également s'il faut améliorer les données de validation, y compris les données expérimentales.

Rédiger un rapport, établi d'après les résultats de la validation, qui montre l'utilisation correcte et appropriée du programme informatique.

Vérifier que le processus de validation a été effectué conformément au plan de validation et que les résultats de la validation sont signalés avec précision.

5.0 MAINTENANCE

5.1 Gestion de la configuration

Établir les procédures relatives à la maintenance du programme informatique et au contrôle des changements.

Adopter des pratiques qui protègent l'intégrité du programme informatique. Définir clairement les rôles et responsabilités des personnes qui sont chargées de veiller à l'intégrité du programme, ainsi que de celles qui en assurent la maintenance et l'utilisent.

Avoir un système de gestion de la configuration qui contribue à garantir l'intégrité du programme informatique et à tenir registre des versions et des éléments du programme.

Au nombre des éléments-types d'un programme informatique, figurent :

- le programme source;
- le programme objet;
- le programme exécutable;
- les processeurs et les systèmes d'exploitation;
- les fichiers servant à contrôler la génération de code exécutable à partir du code source;
- les fichiers de données d'entrée;
- la documentation;
- les outils utilisés à des fins de soutien au développement ou à la maintenance du programme.

La gestion de la configuration porte sur les points suivants :

- utiliser des règles d'affectation des noms pour identifier de façon unique les caractéristiques fonctionnelles et physiques de chaque élément de la configuration;
- contrôler les changements apportés aux éléments de la configuration;
- noter et signaler l'état de la mise en œuvre et du traitement des changements;
- vérifier la conformité avec les exigences relatives à la configuration.

5.2 Contrôle des changements

Établir un système de contrôle des changements qui permette de gérer les changements apportés à chaque élément de la configuration. Le système de contrôle des changements devrait comprendre les éléments suivants :

- une personne désignée qui est responsable de tous les changements apportés au programme informatique;
- un groupe d'approbation des changements, formé de représentants des développeurs, des utilisateurs et des personnes qui procèdent à la vérification et à la validation du programme;
- des procédures pour :
  - examiner les changements proposés pour un élément de la configuration;
  - approuver les changements;
  - mettre en œuvre les changements;
  - documenter les changements et leur raison d'être;
  - s'assurer que les changements ont été mis en œuvre correctement;
  - évaluer l'incidence de chaque changement sur l'utilisation du programme et sur la qualité de ses prédictions;
  - apporter toute révision jugée nécessaire à la documentation déjà établie.

6.0 UTILISATION DES PROGRAMMES INFORMATIQUES

6.1 Aperçu

Établir des lignes directrices et des procédures pour l'utilisation des programmes informatiques. Elles devraient comprendre des lignes directrices sur les préalables exigés de l'utilisateur et sur la désignation des personnes chargées de s'assurer de l'utilisation correcte et appropriée des programmes.

N'appliquer un programme informatique qu'à un problème ou qu'à un ensemble de problèmes pour lequel (lesquels) il a été validé et qu'il vise à résoudre — et ce, seulement si les domaines d'applicabilité identifiés dans le rapport de vérification et de validation peuvent être respectés. Si le programme est utilisé à l'extérieur de ces domaines de validation, la validité de l'extrapolation devrait être justifiée.

Dans la mesure du possible, réduire au minimum les effets du programme informatique attribuables à l'utilisateur. Il y aura lieu de dire que de tels effets se sont effectivement produits si, par exemple, deux utilisateurs expérimentés qui emploient la même version d'un programme effectuent un test ou analysent un problème et obtiennent des résultats assez différents. Ces différences peuvent être attribuables aux options d'entrée qui ont été utilisées — le contrôle des intervalles de temps, les schèmes de nodalisation, les corrélations, les coefficients ou les paramètres retenus, par exemple.

Signaler toute erreur ou lacune du programme informatique ou de son application à la personne responsable de la maintenance de ce programme. Corriger les erreurs et les lacunes en suivant les procédures de contrôle des changements (voir la section 5.2 du présent guide).

6.2 Analyse des incertitudes

Évaluer les résultats du programme informatique pour déterminer les incertitudes rattachées à chacun de ses divers types d'application.

Tenir compte des incertitudes éventuelles relevées lors du développement et de l'utilisation du programme informatique, y compris celles qui résultent :

- des simplifications et hypothèses qui ont été faites pour compenser le manque de connaissances ou pour permettre la solution d'un problème, y compris les hypothèses ayant servi à simplifier les équations, les corrélations empiriques, les modèles physiques et les modèles du système;
- d'une connaissance insuffisante du problème que le programme informatique vise à analyser, à savoir les phénomènes complexes, les effets d'échelle et les conditions initiales et aux limites;
- des limitations inhérentes à la capacité de représenter un système physique — celles qui sont attribuables aux techniques de nodalisation, par exemple;
- des limitations inhérentes aux données servant à valider le programme — celles qui sont attribuables à des erreurs d'instrumentation ou qui tiennent à la répétabilité des essais;
- de la compensation des erreurs ou des incertitudes;
- de la propagation des incertitudes — celles qui sont attribuables au fait que l'incertitude des résultats augmente au fur et à mesure que s'accroît le nombre des phénomènes transitoires.

6.3 Préalables exigés de l'utilisateur

Prendre des mesures pour s'assurer que chaque utilisateur d'un programme informatique possède l'expérience qui correspond au profil de sûreté de l'application, voire qui le dépasse.

Les facteurs importants pour ce qui est des connaissances et de l'expérience exigés de l'utilisateur sont les suivants :

- une bonne connaissance et une bonne compréhension de l'information contenue dans tous les documents d'application et dans le rapport de vérification et de revue de la conception; une connaissance sommaire des domaines desquels relèvent les autres renseignements figurant dans les documents de conception;
- une expérience suffisante de l'utilisation du programme informatique pour l'application prévue, et pour d'autres de même nature, avec une bonne connaissance des réponses du programme (1) à divers modèles de phénomènes et de systèmes et (2) aux hypothèses et changements apportés à des paramètres importants;
- une connaissance étendue du problème à analyser — le système physique et les phénomènes modélisés qui s'y rattachent, par exemple.

6.4 Soutien de l'utilisateur

Mettre à la disposition de l'utilisateur des services de soutien qui permettront de le conseiller sur l'utilisation correcte du programme informatique et de modélisations ou simulations appropriées lorsqu'il est aux prises avec des programmes longs et complexes — ceux qui portent sur les systèmes thermohydrauliques, ou encore qui exigent le couplage de plusieurs champs de données, par exemple.

Mettre sur pied, le cas échéant, une équipe de soutien de l'utilisateur, qui sera formée des membres suivants :

- le développeur ou des personnes qui connaissent bien le programme informatique;
- des personnes qui ont procédé à la vérification et à la validation du programme ou qui connaissent bien ces deux processus;
- des personnes qui ont une connaissance approfondie du système à modéliser et du problème à résoudre.

6.5 Options de l'utilisateur

Éviter, dans toute la mesure du possible, que l'utilisateur ait à choisir entre diverses options d'entrée.

Rédiger des lignes directrices pour chaque option de l'utilisateur et les rendre accessibles à tous les utilisateurs.

6.6 Vérification du processus d'application

S'assurer que le processus d'application est correct en démontrant qu'il satisfait aux lignes directrices sur l'utilisation des programmes informatiques énoncées dans l'ensemble de la présente section 6.

Si le processus d'application ne peut être vérifié de manière satisfaisante, identifier les exceptions et fournir une justification pour chacune d'elles.

7.0 DOCUMENTATION

7.1 Aperçu

La documentation du programme informatique et des éléments connexes devrait être conforme aux normes de documentation adoptées pour le développement du programme. Effectuer une vérification des documents du programme pour démontrer qu'ils sont complets, cohérents, clairs et sans ambiguïté.

7.2 Création de la documentation de développement

Dans la documentation de développement, inclure les éléments suivants :

- la définition du problème;
- le plan de développement du programme;
- la spécification des exigences;
- la spécification des exigences de conception;
- le manuel du programmeur;
- le rapport de vérification et de revue de la conception.

Dans le document traitant de la définition du problème, documenter en détail l'énoncé du problème, de même que la raison d'être et les objectifs du programme informatique.

Dans le document traitant du plan de développement du programme informatique, décrire l'organisation, l'échéancier et les activités liées au développement, à la revue de la conception, à la vérification et à la validation du programme. Documenter également les procédures de mise à jour du plan de développement du programme.

Dans le document traitant de la spécification des exigences, identifier clairement toutes les exigences du programme informatique. Faire, en outre, état des éléments de base servant à vérifier la conception du programme et à évaluer, grâce au processus de validation, la performance du programme.

Dans le document traitant de la spécification des exigences de conception, préciser les éléments qui sont nécessaires au codage du programme informatique, dont la structure logique, le flux de l'information, les modèles, les techniques de solution numériques, la méthode de discrétisation, la structure des données, le matériel et les logiciels de soutien et l'environnement opérationnel.

Dans le manuel du programmeur, décrire le déroulement et la structure du programme informatique, la méthode utilisée pour traduire la théorie en codage, les instructions relatives à la maintenance et à la modification du programme et les conventions quant aux pratiques de programmation — la façon de nommer les variables et de consigner des observations dans le programme, par exemple.

Dans le rapport de vérification et de revue de la conception, documenter les activités de vérification, en faisant notamment état de la méthode et des résultats de chaque vérification ou revue de la conception, ainsi que des domaines d'applicabilité du programme informatique et des recommandations destinées aux utilisateurs concernant ses capacités et ses limitations. Mentionner également les données d'entrée utilisées pour chaque vérification.

7.3 Documentation d'application

La documentation d'application devrait comprendre les éléments suivants :

- un sommaire du programme informatique;
- un manuel de théorie;
- un manuel de l'utilisateur comportant notamment des lignes directrices destinées à des utilisateurs particuliers;
- un ou plusieurs rapports de validation;
- le code source;
- des échantillons d'entrées et de sorties.

Dans le sommaire du programme informatique, fournir les renseignements suivants au sujet du programme :

- le nom et la version du programme, ainsi que les éléments de configuration pertinents;
- l'objet, les capacités, les limitations et l'environnement opérationnel du programme;

- un sommaire du ou des problèmes que le programme vise à résoudre;
- le nom de l'organisation et des personnes clés responsables du développement du code, ainsi que du soutien et de la maintenance du programme.

Dans le manuel de théorie, décrire les éléments de base du programme informatique, dont, entre autres :

- les systèmes physiques à modéliser et leurs modèles;
- les phénomènes à modéliser et leurs modèles ou corrélations empiriques;
- les formulations mathématiques du problème et les techniques de solution;
- les hypothèses et les contraintes, de même que leurs incidences sur les capacités du programme, sur ses limitations et sur ses domaines d'applicabilité;
- la liste des documents de référence.

Dans le manuel de l'utilisateur, faire état des renseignements qui permettront d'exécuter le programme informatique. Fournir des détails sur les données d'entrée qui sont exigées, sur les techniques de redémarrage du code, sur les méthodes de diagnostic et sur les options de l'utilisateur, y compris les éléments suivants :

- l'information de base sur la compilation, l'enchaînement et l'exécution du code;
- la description détaillée de tous les paramètres relatifs aux entrées, en indiquant le type et le format voulus;
- une discussion sur le pré-traitement et le post-traitement du code, y compris sur son redémarrage;
- des échantillons de fichiers de données d'entrée et de sortie qui montrent des problèmes représentatifs;
- une description des messages faisant état d'erreurs d'exécution et de la fin de l'exécution du programme.

Pour réduire au minimum le risque d'une utilisation non appropriée du programme informatique, fournir en outre à l'utilisateur des lignes directrices portant sur les sujets suivants :

- l'utilisation correcte et appropriée du programme pour chacune de ses applications;
- les domaines d'applicabilité du programme;
- les capacités et les limitations du programme;
- les options de l'utilisateur, en matière de modèles ou de schèmes de nodalisation notamment.

Dans le rapport de validation, inclure les éléments suivants :

- le plan de validation;
- les résultats de la validation;
- une évaluation des résultats de la validation, où il est notamment fait état de l'exactitude du programme informatique et de la gamme d'incertitudes qui y sont rattachées pour chacun des essais;
- des recommandations quant à l'amélioration du programme;
- des précisions quant à des expériences ultérieures;
- des recommandations quant à l'utilisation correcte et appropriée du programme pour chacune de ses applications, quant à ses domaines d'applicabilité et quant à ses capacités et limitations.

8.0 PROGRAMMES INFORMATIQUES EXISTANTS

Les programmes informatiques existants sont ceux qui étaient utilisés par le titulaire de permis de la CCSN avant l'entrée en vigueur du présent guide.

Évaluer dans quelle mesure les programmes existants sont conformes au présent guide et documenter ces évaluations.

Améliorer, dans la mesure du possible, les domaines qui ne sont pas conformes aux sections pertinentes du présent guide pour faire en sorte que le niveau de confiance envers les programmes informatiques existants soit concomitant à leur incidence sur la sûreté.

Fournir une justification pour tout cas de non-conformité aux lignes directrices du présent guide.

9.0 ACQUISITION DE PROGRAMMES INFORMATIQUES

Établir des procédures pour l'acquisition et la maintenance des programmes informatiques. Veiller à ce que tous les programmes qui, bien qu'élaborés à l'extérieur de l'organisation du titulaire de permis de la CCSN, seront utilisés par le titulaire de permis lui-même satisfassent aux critères du présent guide, voire qu'ils les dépassent.

GLOSSAIRE

codage
Étape au cours de laquelle la conception du programme informatique est transformée en une séquence d'instructions lisibles par machine pouvant être traitées par un ordinateur.

conception du programme informatique
Éléments nécessaires pour le codage du programme informatique, y compris la structure logique, le flux de l'information, les modèles, les techniques de solution numériques, la méthode de discrétisation, la structure des données, le matériel et les logiciels de soutien et l'environnement opérationnel, ainsi que d'autres caractéristiques permettant de satisfaire à la spécification des exigences.

définition du problème
Description détaillée du problème et de la raison qui justifie la décision de développer le programme informatique visant à le résoudre.

développeur
Personne qui rédige le code d'un programme informatique, ou une partie de ce code, et qui participe à l'élaboration des documents connexes.

discrétisation
Méthode d'approximation de la fonction mathématique réelle destinée à être intégrée.

intégration du programme
Intégration du programme source à d'autres éléments, comme les programmes de bibliothèque, la spécification des liens du système, le langage de commande du système d'exploitation, les bibliothèques de données externes et l'environnement matériel, de manière à former un programme exécutable intégré.

programme informatique
Séquence d'instructions lisibles par machine pouvant être traitées par un ordinateur, également parfois désignée sous le simple nom de « programme » dans le présent guide.

programme source
Programme qui doit être compilé, assemblé ou interprété avant d'être exécuté.

spécification des exigences
Description des exigences auxquelles le programme informatique doit satisfaire, lesquelles serviront par la suite de base à la vérification de la conception du programme et à l'évaluation du programme jusqu'à sa validation.

utilisateur
Personne qui utilise le programme informatique.

validation
Analyse comparative des résultats du programme informatique, d'une part, et des mesures opérationnelles, données expérimentales ou solutions analytiques, d'autre part, afin de déterminer le degré d'exactitude et d'incertitude du programme.

vérification
Analyse servant à déterminer si le produit d'une étape donnée du cycle de développement du programme informatique correspond à la spécification des exigences établie pendant l'étape précédente.